RSO Characterization with Photometric Data Using Machine Learning

Michael Howard
Charles River Analytics, Inc.
Bernie Klem
SASSO, Inc.
Joe Gorman
Charles River Analytics, Inc.

1. ABSTRACT

Object characterization is the description of a resident space object (RSO), its capabilities, and its behavior. This paper explores object characterization methods using photometric data. An important property of RSO photometric signatures is the changes in intensity that they exhibit with respect to changes in viewing angle or orientation. Properties that influence the brightness of the photometric signature include geometry, orientation, material characteristics and stability. For this reason, it should be possible to recover these characteristics by analyzing photometric signatures. In this paper, we discuss the application of machine learning techniques to RSO characterization. We develop simulated signatures in the visible band of three basic RSO types, with variations in object orientation, material characteristics, size and attitude. We generate observations by sampling measurements from the simulated signature. Next, we apply a feature extraction technique to the simulated signatures and train a variety of machine learning algorithms to classify the signatures. The classifications are made on sequences of one or more measurements. We consider the effectiveness of a set of binary classifiers trained to recognize one of each case. The results of each classifier are combined together to produce a final output characterization of an input observation. Experiments with varying levels of noise are presented, and we evaluate models with respect to classification accuracy and other criteria.

2. MOTIVATION

Object characterization is the description of a resident space object (RSO), its capabilities, and its behavior. An important property of RSO photometric signatures is the changes in intensity that they exhibit with respect to changes in viewing angle or orientation. Properties that influence the brightness of the photometric signature include geometry, orientation, material characteristics, and stability. For this reason, it should be possible to recover these characteristics by analyzing photometric signatures. However, space analysts currently perform most RSO characterization manually and on an individual basis, and there is a need for efficient and automated methods to perform characterization. In particular, we have focused on object characterization in the geosynchronous regime.

This paper discusses a system called Characterizing Resident Space Objects with Photometric Signatures (CROPS). We develop simulated signatures in the visible spectral band of three basic RSO types, with variations in object orientation, material characteristics, size and attitude and attempt to recover these properties through machine learning object characterization techniques. Machine learning is a technique for building models to perform tasks like object characterization. Specifically, a classification model takes an input and categorizes it. We develop a set of classifiers capable of providing space operators with fast, accurate characterizations of RSOs, and consider the performance envelope with respect to noise and training data availability.

3. SIMULATED PHOTOMETRIC DATA

To demonstrate the feasibility of describing space objects based on their photometric signatures, we created simulated signatures for a set of space object types. The illustrations here depict an approximate representation of the subject space objects. These approximations are more than sufficient for us to capture the effects of the major signature producing components and observable features needed to demonstrate the capability and success of our machine learning processes. Adding additional geometry details would only increase the accuracy of the

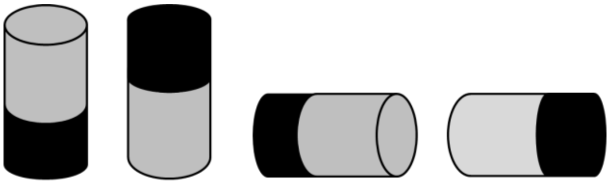

Fig. 1 - Rocket Body
(A) Dark nadir (B) Dark anti nadir
(C) dark left (D) dark right

object details but would add little to affect the overall positive achievement of the results presented in Section 6. This also emphasizes the utility of applying streamlined signature modeling to demonstrate machine learning capability. First, we determined a set of space objects to produce simulations of space objects. We chose to simulate the signatures of three object types: A communication satellite, based on the Galaxy 15 architecture; a cube satellite, and a cylindrical rocket body. Simulations for the communication satellite featured both a cube-shaped bus and a cylindrical bus. Thus, there are enough similarities that identification of the cases is not trivial.

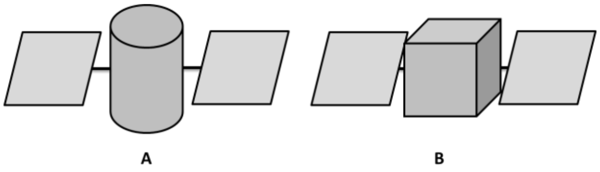

Fig. 2 – Communication Satellite
(A) Cylindrical Central Body (B) Cubic Central Body

Additionally, we include signatures of both stable and tumbling objects. The modeled objects contain some surface non-uniformities in geometry and material characteristics, and multiple stable orientations and attitudes. In some cases, object size is also varied. The signature predictions consist of walk around views for stable objects and single views for tumbling objects for a single solar incidence scenario. Most geosynchronous payloads will be body nadir pointing and therefore one does not expect to see a second face come fully into view. Later, when we use classification techniques to identify the object in these signatures, we create 'observations' of the objects by extracting points from a subsection of the signature and apply random noise to the points. It is important to note that the classification models operate on the points from a subsection and not the full signature. Each simulation is modeled in the visible band. In these simulations, the broadening of the solar glints and absolute glint magnitudes are not intended to be high fidelity. However, the relative magnitudes and location of the glints on the angle scale are accurate. In total, there are 15 cases of the three space object types.

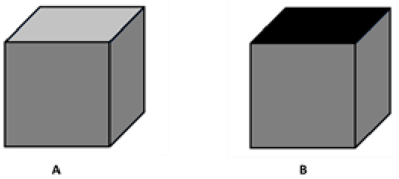

Fig. 3 – CubeSat
(A) Dark Nadir (B) Dark Anti-nadir

We produced simulations for two variations of a communication satellite, modeled after the Galaxy 15. One case features a cubic central body, and the other a right-circular cylinder central body. Fig. 2 is an illustration of the cases. Fig. 3 shows the simulated signatures of the stable cases on the left, and tumbling cases on the right. A visual inspection indicates that the signatures of the central body types are quite similar, with the most significant differences in magnitude appearing at a viewing angle around 50 degrees. For the stable cases, there are four glints per cycle. In the tumbling case, eight glints appear in each cycle.

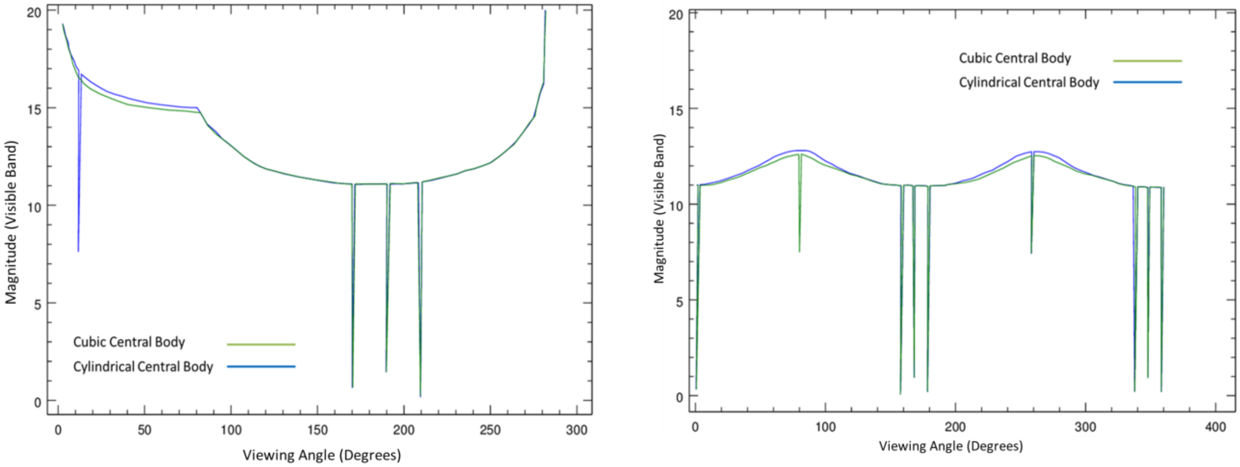

Fig. 4 – Communication Satellite Signatures

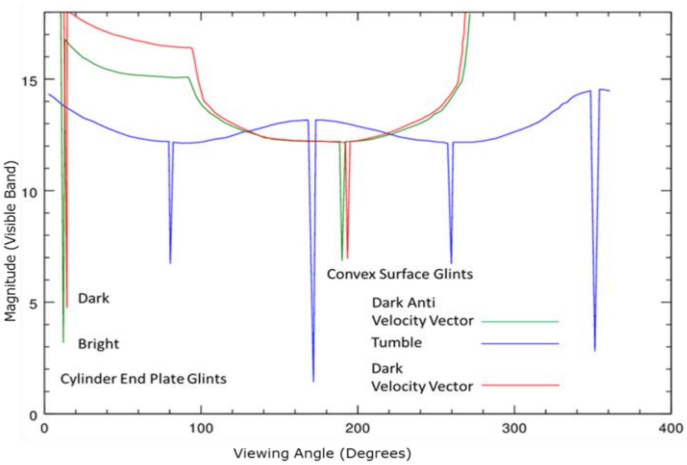

Fig. 5 – Vertical Rocket Body Signatures

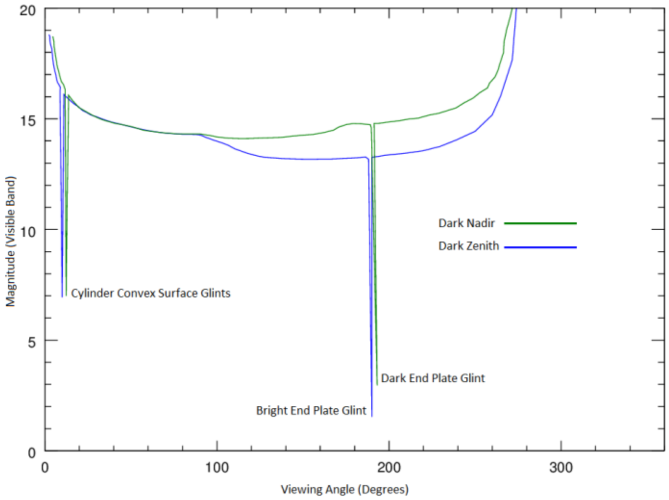

Fig. 6 – Horizontal Rocket Body Signatures

The rocket body was modeled as a cylindrical object with one end featuring dark paint. We produced six stable cases of the rocket body: Stable, with the dark end at the nadir and at zenith, as well as with the dark side facing the velocity vector and anti-velocity vector. We also included a tumbling case. Fig. 1 shows the different orientation cases included in our simulations.

Fig. 6 and Fig. 5 show the simulated signatures for the rocket body in both orientations. For the vertical stable cases, there are two glints per cycle. The first is the convex surface glint. The second is the end plate glints. Note that there is a difference in the relative magnitudes of these glints, due to the bright or dark color at the end of rocket body. In the horizontally oriented stable case, there are similarly two glints per cycle, with the first being the end plate glint, which again exhibits the difference in relative magnitude.

In the tumbling case, there are five glints per cycle. The brightest glint is the bright end plate, the second brightest is the dark end, and the remaining three are the convex surface glints. We modeled a cube satellite featuring four sides with solar panels, one dark side and one bright side (see). The cube satellite is modeled at two sizes – 1m and 10cm. We produced simulations in which the dark side is at the nadir and zenith, as well as a tumbling case.

Fig. 7 contains the simulated signatures of all six cases. Note that slight rotations in azimuth (3 degrees) would produce a similar signature profile, but without the glints. The stable case features a glint from the solar panel side surface, and a glint from the bright or dark side. The relative magnitude of this glint is a distinguishing characteristic of the two cases. In the tumbling case, there are four glints per cycle. Two glints from the side surfaces, and one each from the bright and dark top and bottom. In all cases, the 10cm cube is about five magnitudes fainter than the 1m cube.

The next step is to define the inputs to the classification model. To create a classification model from our 15 simulated signatures, we need to create a large set of observations. By *signature*, we refer to the continuous simulated curve described in the preceding section. By *observation*, we mean a set of points representing the measured magnitudes resulting from observing a space object. Clearly, real-world observations will not be a perfectly smooth curve and will contain a limited number of points. A method must be devised for extracting observations from our simulated signatures. Additionally, because of uncertainties in the viewing conditions, seasonal effects, or calibration of viewing equipment, we also need to account for possible noise in the observations. To help

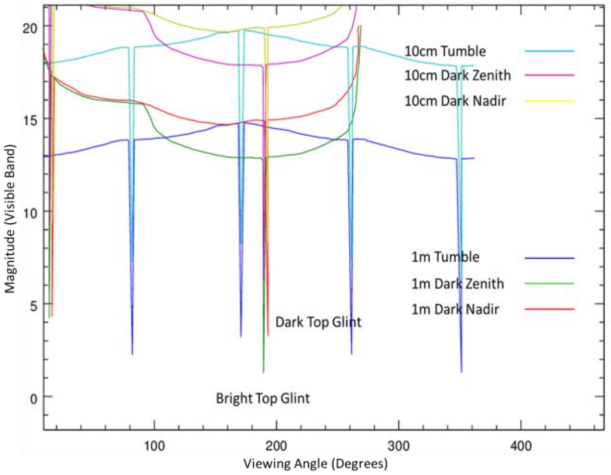

Fig. 7 - CubeSat Signatures

understand how to accomplish this, we considered the schema for the GEODDS system and examined the data present in the Geocolor Photometry Catalog. For both cases, an observation consists of a set of measurements with an interval of time separating each measurement. Thus, to create these types of observations from our simulations, we first choose an initial angle θ and a number of points, n. We take a section of a simulated curve, then sample a value at θ and advance the viewing angle by a small, uniformly distributed amount. This is repeated n times. Optionally, when each value is sampled, we generate a small, normally distributed noise term and multiply it by the sampled value. In other words, for each sampled measurement m, we create a new measurement by multiplying by normally distributed random noise:

$$m_{new} = am, a \sim Normal(1.0, \sigma^2)$$

This sampling process is repeated to generate several thousand observations from each simulated signature, and each of the resulting observations are distinct. Note that the number of points and the amount of noise present is a parameter of the process.

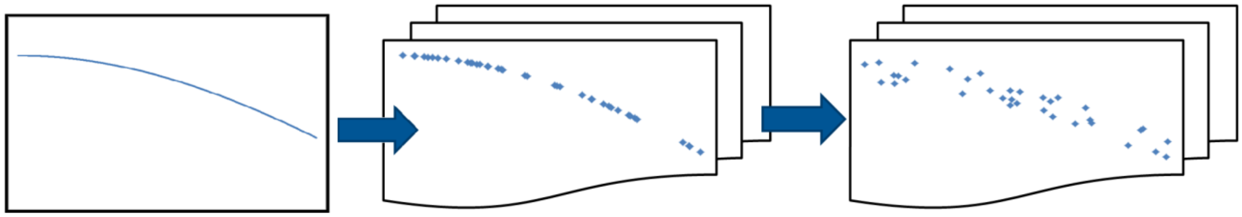

Fig. 8: From observation to noisy observation

In most of our experiments, we used a variance of $\sigma^2 = 0.05$. In section 7, we report the results of changing this parameter as well as reducing the number of measurements contained in each observation. As an illustration, Fig. 8 shows a section of $\cos(x)$ before and after sampling, and after multiplication by a random noise term. This is a simple approach to simulate noise from a variety of potential sources.

4. FEATURIZATION

The next step is to convert each observation in the dataset into a representative set of features. There are well known observable features which affect the photometric signature of a space object: size, motion period, location and frequency of glints, relative peak magnitudes of glints, possible ratio of maximum to minimum diffuse signatures, and absolute and relative magnitudes of diffuse signatures all impact the shape of a signature. The features in Table 1 transform an input observation from a set of photometric measurements into a consistent set of qualities for comparison to separate different classes of objects. Note that data from any source that produces photometric measurements can be transformed into the CROPS feature set.

Table 1: Representative features

	Feature	Description	Algorithm
Dominant frequency features	Dominant frequency	Finds the *N* largest *FFT* bins	$DF = i, j, k \quad where \begin{cases} FFT_i > FFT_j > FFT_k \\ \forall_{l \in FFT} (FFT_i > FFT_l) \\ \forall_{l \in \{FFT - FFT_i\}} (FFT_j > FFT_l) \\ \forall_{l \in \{FFT - FFT_i - FFT_j\}} (FFT_k > FFT_l) \end{cases}$
	Peak support value	Computes the number of frequency samples that slope upward to the peak on both sides	$PS_i = N_{ascending} + N_{descending}$
	Inter-peak distance	Computes the difference in frequency between subsequent peaks	$PD_i = DF_{i+1} - DF_i$ (where the DF_i have been ordered by frequency)
Mel-frequency Cepstral coefficients	M0-M10	Applies a set of triangular filters designed to isolate specific patterns, but tuned to the photometric intensity range	$c(n) = \frac{1}{N} \sum_{k=0}^{N} \log \| FFT_{avg}(k) \| e^{j 2 \pi k \frac{n}{N}}$
Spectral centroid	Sc	Computes center of mass of the *FFT*	$Sc = \sum_{i=0}^{N} i * FFT_i / \sum_{j=0}^{N} FFT_j$
Spectral Ratio	Sr	Computes the weighted sum of the lower *N* frequency bins to the upper bins	$Sr = \int_{x0}^{x1} \| X(f) \| df / \int_{x1}^{x2} \| X(f) \| df$

The aim is to capture the distinguishing characteristics of each space object type and status. For instance, the dominant frequency feature identifies the location of glints, their frequency, and their relative peak magnitudes.

There are a considerable number of free parameters in the feature set. The presence of a single feature can be indicative of a specific capability of space object type. The features serve as the basis of data exploitation, and are thus the key to object classification and characterization. In addition, these features are agnostic to the source of the observations and could be extracted from multiple photometric bands or sources producing photometric data.

5. CLASSIFICATION MODELS

The data exploitation component contains a set of machine learning models to infer the type and status of space objects. The type of machine learning task that the CROPS data exploitation models apply is called classification. Classification means that the model is presented with one instance of data – in our case, an observation represented by the feature set described in section 4– and must assign it a classification label from a discrete set of choices. The classification label is the type of space object that the model believes is present in the observation. The results of transforming observations to representative features are input to a set of four classifiers, each trained to recognize one of the three basic space object types in our dataset, as well as the tumbling status of the object. The CROPS data exploitation component contains a separate classification model for each space object type. We refer to these first four models as the 'first tier classifiers.' The reasoning behind this approach is that certain machine learning techniques may be more apt at identifying particular object types. Additionally, we may want to include or exclude specific features for identifying space object types that are especially similar or difficult to discern. The architecture is flexible enough to enable more complicated data exploitation scenarios in the future when classifying a large number of space object types may be required.

For each input observation, the output of each model is a classification label and a probability. The probability represents the confidence or belief of the classifier in the label that it has chosen. The models participate in the overall object classification in a 'one-against-all' fashion, meaning that we have a separate model for recognizing each type of object and each is trained to recognize only one specific type of object. For instance, when we train a model to identify a communication satellite, the dataset includes observations from all four simulations. This includes stable and tumbling rocket bodies and communication satellites.

Each model outputs a probability that the given observation is a match to the space object type or status that the model has been trained to recognize. If the probability is above .50, then the output is considered as a 'yes' or a 'positive.' As an example, if an observation of a stable communication satellite with a cylindrical central body was input to the classifier, we might see the outputs illustrated in Table 2

Table 2 – Example Classifier Output

	Yes	No
Stable	0.80	0.20
Cube	0.05	0.95
Comm Sat	0.80	0.20
Rocket	0.40	0.60

The output of the classifier with the highest confidence is passed on to its corresponding second tier of classification models, which determine the specific orientation, size or variant of the space object type. While the first tier classifiers were trained using the entire range of space object types, a second tier classifier was trained exclusively on cubesats, communication satellites or rocket bodies. These models are specialized to detect the finer grained characteristics of the space objects. For example, if the first tier of classification models identified an observation as a communication satellite, the second tier 'satellite details' classifier determined whether the satellite had a cylindrical or cubic central body. If the first tier of classification models detected a rocket body, then the second tier determined its orientation, and so on. The entire set of outputs from all classifiers was combined into a single output belief vector containing the highest probability classification, and the probability distribution from each individual classifier.

6. EVALUATION CRITERIA

We applied several standard performance measures to evaluate the performance of the CROPs data exploitation component. Specifically, we examined the accuracy, precision, and recall of each classifier. Each of these metrics are based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) in the output classification. If the classifier assigns a value of '1' to a data instance, and the correct value is '1,' then this instance is counted as a true positive.

Accuracy is an intuitive measure of performance. It is simply the number of correctly labelled instances divided by the total number of instances. It is a value between 0 and 1 – the percent of the data that is correctly classified. *Precision* is the rate at which data instances identified by the classifier as positives are actually positive. *Recall* is the probability that a positive data instance is correctly identified. There are scenarios in which each of these measures might be more valuable than the others. For example, in the case that there are far fewer positive data instances than negatives, a classifier would be able to achieve a high accuracy, even without correctly identifying any positives. In this scenario, one would want to emphasize recall as more important than accuracy or precision. There are metrics that help to find a good balance between accuracy, recall, and precision. F-Measure defines a parameter, β, which controls the degree to which recall is favored over precision. In our experiments, we used $\beta = 1$, meaning we want to balance recall and precision equally.

We examined the performance of a variety of machine learning techniques, including decision trees, logistic regression, neural networks, support vector machines, nearest-neighbors classification, and rule based classification. We applied the models against a dataset of simulated observations and compared their performance. In total, we generated 6000 simulated observations and divided the total in half. The first half of the data was used for training the models, and the second half was used for evaluation. In section 7, we describe our experiments with reducing the number of training instances. We also considered the effects of the noise level in the data set by varying the noise applied to each observation, and considered the outcome of reducing the number of measurements present in each signature.

We examined the performance of a variety of machine learning techniques, including decision trees, logistic regression, neural networks, support vector machines, nearest-neighbors classification and rule based classification. We applied the models against a dataset of simulated observations and compared their performance. While we evaluated a number of different machine learning models, we restricted the data exploitation component to use the same type of classification model for each tier of classification and for each space object type. In future work, we could explore optimization of the data exploitation component by incorporating different classifier types for detecting specific space objects, as well as combining multiple classifiers into an ensemble .

7. RESULTS

Table 3 shows the results of running the data exploitation component on the dataset, using a noise multiplier with 0.05 variance. Training of Random Forest [2][4] Logistic Regression [5], K-Nearest Neighbors [7] and Naïve Bayes [6] classifiers required very little time, and the models were trained after a few seconds. Neural Networks [1] and Support Vector Machines [3] are considerably more sensitive to parameters and required a longer period of time to train. For the full 3000-point dataset, these classifiers were trained after approximately a half hour. After training, however, all classifiers produced their output characterizations nearly instantaneously.

Table 3 – Performance Comparison of Model Types

	Accuracy	Precision	Recall	F1
Naïve Bayes	0.9756	0.9934	0.9577	0.9752
Logistic Regression	0.9316	0.9531	0.9083	0.9302
Random Forest	0.9956	0.9976	0.99363	0.9956
Support Vector Machine	0.9860	0.9880	0.9841	0.9860
Neural Network	0.9856	0.9927	0.9785	0.9855
K Nearest Neighbors	0.9516	0.9263	0.9817	0.9532
CN2 Rules	0.9896	0.9850	0.9944	0.9897

All of the classification models performed at a high level with regard to the evaluation criteria defined in the previous section. The Random Decision Forest (RDF) classifier exhibits the highest accuracy. The classification results of all 15 simulated cases for RDF are shown in a contingency table in Table 5. The keys to the entries of the contingency table are in Table 4. Each cell of the table represents the number of cases. The columns represent the ground truth, and the rows represent the output from the model. For example, the value in row N, column O is '4'. This indicates that a tumbling 10cm cube satellite was misclassified as stable in four cases. A perfect result would contain only entries on the diagonal.

Table 4: Contingency table key

Key	Class	Detail	Status
A	G15	Cylindrical Bus	Stable
B	G15	Cylindrical Bus	Tumble
C	G15	Cubic Bus	Stable
D	G15	Cubic Bus	Tumble
E	Rocket Body	Dark Anti Velocity Vector	Stable
F	Rocket Body	Dark Velocity Vector	Stable
G	Rocket Body	Dark Nadir	Stable
H	Rocket Body	Dark Zenith	Stable
I	Rocket Body		Tumble
J	1m Cube	Dark Zenith	Stable
K	1m Cube	Dark Nadir	Stable
L	1m Cube		Tumble
M	10cm Cube	Dark Zenith	Stable
N	10cm Cube	Dark Nadir	Stable
O	10cm Cube		Tumble

Note that a high number of communication satellite cases were included because it was considered the most difficult case to classify. Also, there are an approximately equal number of tumbling and stable cases of each object type, even if there are multiple stable orientations of the object. The training and testing data were split randomly after the observations were generated and so the representation of each object type and status is approximately even, though not exact.

Table 5: Contingency table of RDF classification results

Predicted \ Actual	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	401	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	404	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	410	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	426	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	102	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	107	1	1	0	0	0	0	0	0	0
G	0	0	0	0	0	0	105	2	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	426	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	123	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	113	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	63	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	2	84	0	0
N	0	0	0	0	0	0	0	0	1	0	0	0	0	70	4
O	0	0	0	0	0	0	0	0	0	0	0	0	0	1	59

These results established the feasibility of characterizing space objects from photometric data. However, a number of concerns remain. Because of uncertainties in viewing conditions, seasonal effects, and instrument calibration, there may be considerable noise present in observations. Second, the noise level present in deployment may be greater than the noise level used to train the model, or have a different distribution. Third, the amount of real-world measurements available may be considerably scarcer than what was used in our sampling process. Finally, the computational cost of characterizing an observation should be as low as possible, and the required time to update or train the model should also be tractable.

We can restate these considerations in terms of common machine learning concepts. First, it must be ensured that the classification technique employed by CROPS is resistant to over fitting. Over fit means that the learned model can only recognize cases which are identical or very similar to the cases contained in the training data. We assessed the resistance to over fitting by testing against a different noise level and sampling rate than what was used in the training data, comparing the results across the different classifiers, and selecting a technique which performs well. We refer to this as a change in environment, meaning that the conditions of the training environment are not necessarily the same as the deployment environment. Second, the CROPS exploitation component must be resistant to noise. CROPS should be able to provide useful characterizations of space objects, even in noisy conditions. We evaluated the robustness to noise by varying the amount of noise in the data. Lastly, we measured the performance of the model using a smaller number of measurements per observation to investigate how the accuracy degrades as data becomes increasingly scarce.

To evaluate resistance to over fit, we trained the models using the original dataset, but tested them against a dataset in which the variance of the noise multiplier is increased from 0.05 to 0.15. The results are shown in Table 6. Clearly, certain classifiers are more robust to this type of change in environment type of data than others. The results suggest that in deployment, CROPS should use a nonlinear classifier such as Random Decision Forests, Support Vector Machines or Neural Networks.

Table 6: Comparison of classifier performance on noisy test data

	Accuracy	Precision	Recall	F1
Naïve Bayes	0.662	0.652	0.495	0.9507
Logistic Regression	0.7989	0.6622	0.7478	0.5942
Random Forest	0.8604	0.762	0.8768	0.6737
Support Vector Machine	0.854	0.7662	0.8174	0.721
Neural Network	0.7692	0.6251	0.6776	0.5801
CN2 Rules	0.8417	0.7349	0.8264	0.6616
K Nearest Neighbors	0.6416	0.4605	0.4598	0.461

Specifically, we found that the best choice is the Random Decision Forest. RDFs share many of the desirable characteristics that make non-random decision trees popular: they are computationally extremely efficient (they run in constant time), and they can find discriminative subspaces in high dimensions. Importantly, RDFs are specifically intended to avoid over-fitting that commonly occurs in other model types.

The objective of space object characterization is to inform space operators and analysts. The characterization of a space object can inform decisions about actions which need to be taken. An inconclusive characterization might motivate an operator to schedule additional observations of the same object. It is the goal of this section to determine how well our classifiers are able to characterize objects under different noise conditions and with more scarce amounts of observations. We may infer that substantial information can by gained through additional observations. Or, we may conclude that we have already learned as much as possible from the observations already collected, and that there is little value in re-observing the object from a characterization standpoint. By showing the effectiveness of our system under these different conditions, we provide not only the space object characterizations, but also context about the fidelity of the system and its normal operation.

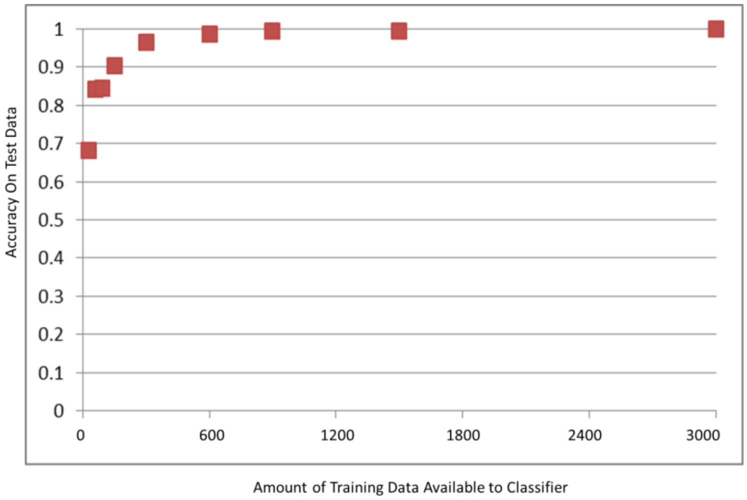

Fig. 9 - RDF accuracy versus training set size

We explored the behavior and context of our classification system using RDF classifiers. To accomplish this, we performed a number of classification experiments under varying conditions and compared the differences. Specifically, we can view the performance of the system under the different conditions as a learning curve and examine its properties. Through creation of data with different noise levels, we have made a rough approximation of different viewing conditions, seasonal effects, sensor biases, and other sources of uncertainty. This introduces a few new notions for consideration: the *initial accuracy*, which is the test score using the minimum number of training instances. The *learning rate*, which is the rate of improvement in test accuracy as the number of training instances increases. The *asymptotic accuracy* is the accuracy a classifier could expect to obtain after seeing infinitely many training instances.

Fig. 9 shows the accuracy achieved on the test dataset when using smaller percentages of the available training data. The test data is the same at every point in the curve. What has changed is the amount of data which the classifier is trained against. We can examine the improvement in performance when the classifier is allowed to train with more samples. The figure shows that the performance using 5% or 10% of the training data is close to the performance using 100% of the data. In other words, the classifier has essentially reached its asymptotic accuracy using a training size much smaller than the original 3000 point training set. Further, while there is considerable performance degradation when the classifier is trained with less than 5% of the training data, even the initial accuracy obtained with only 1% of the full training data is still above the expected accuracy from a random guess. It is apparent that the RDF classifier can produce a meaningful result even with a 1% of the total training data (30 training cases). The classifier achieves its asymptotic accuracy after 10% of the full training set (300 cases).

Finally, to we varied the number of *measurements* present in each observation. In our original experiment, we sampled between 40 and 100 measurements per observation. We also increased the noise level from 0.05 to 0.10 and from 0.10 to 0.15. In Fig. 10 and Fig. 11, we show the results of restricting the number of observations to be between 1 and 10. The classification technique in each plot is again the RDF classifier. While the performance is degraded, the data exploitation component is still making useful characterizations. Because there are 15 possible outputs, the expected performance of a classifier making random guesses is 1/15 ≈ 0.06. Although our model's performance was reduced to about 40%, this is still a much higher level of performance than the baseline. Even the initial accuracy, meaning the accuracy obtained using a very small training set, is still above the baseline.

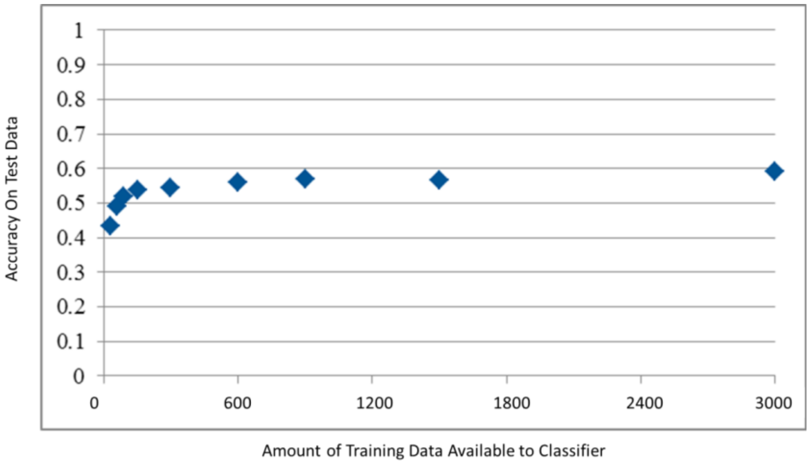

Fig. 10 - RDF performance envelope with medium noise (0.10) and restructured measurements per observation

Besides the difference in initial and asymptotic accuracy, the learning rate is also slower when the noise variance is increased. When the noise variance is increased to 0.10, the classification system still has a moderate accuracy improvement between a training size of 30% (900 data instances) and 40% (1200 data instances). It is also interesting to compare the difference between the initial and asymptotic accuracy. This is an indication of the amount of improvement that can be made if more data is available to the classifier. By comparing Fig. 10 with Fig. 11, it is evident that there is less total improvement that can be gained over the initial accuracy under higher noise levels. One solution to obtaining better accuracy in noisy conditions would be to revisit the feature set in and develop more informative feature extraction methods. Another approach might be to apply a de-noising procedure before transforming the measurements into features.

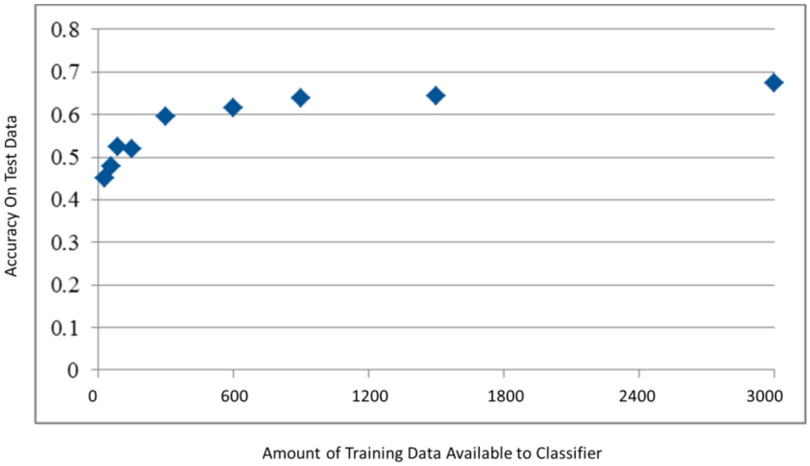

Fig. 11 - RDF performance envelope with high noise (0.15) and restructured measurements per observation

These results are significant, because we have shown that CROPS can produce accurate characterizations with a minimal set of training observations: a small number of measurements per observation, even when a considerable amount of noise is present in the data. Additionally, because the classifiers report their confidence in their output, these results demonstrate that if the correct characterization of an object is uncertain, it can be made more accurate by providing more observations of the object.

8. SUMMARY

We have presented a novel application of machine learning to the task of classifying photometric signatures. There are multiple avenues to expand and improve upon the material presented, including by richer simulations, inclusion of multiple photometric bands, inclusion of real-world satellite data, extraction of more robust features, and application of a more sophisticated machine learning architecture. The results of this research could have significant applicability to a number of SSA activities, including lost track recovery, catalog maintenance, and change detection.

9. REFERENCES

1. Dreiseitl, S. and Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35, 352-359.
2. Fan, W., Wang, H., Yu, P. S., and Ma, S. (2003). Is Random Model Better? On its Accuracy and Efficiency. In *Third IEEE International Conference on Data Mining*, 51.
3. Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, 14.
4. Ho, T. K. (1995). Random Decision Forests. In *IEEE International Conference on Document Analysis and Recognition*, 278-282.
5. Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2000). *Introduction to the logistic regression model*.: Wiley Online Library.
6. Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." *Machine learning: ECML-98*. Springer Berlin Heidelberg, 1998. 4-15.
7. Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." *Neural Networks, IEEE Transactions on* 16.3 (2005): 645-678

10. ACKNOWLEDGEMENT

This material is based upon work supported by the United States Air Force under Contract No. FA9453-14-M-0153.